CLAIMS

What is claimed is:

1. A method comprising:

receiving file system data;

storing the file system data in a plurality of reserved sectors within a non-volatile memory;

compressing the file system data stored within in the plurality of reserved sectors to create a compressed data block; and

storing the compressed data block in at least one physical subsector within the non-volatile memory.

2. The method as recited in Claim 1, wherein receiving file system data further includes presenting an operating system with a plurality of operatively accessible virtual sectors resulting in a virtual memory capacity that exceeds the actual physical capacity of the non-volatile memory.

3. The method as recited in Claim 2, wherein storing the compressed data block at least one physical subsector within the non-volatile memory further includes mapping the plurality of virtual sectors to at least one physical subsector through a Virtual Sector Table (VST) stored in a volatile memory and presenting the operating system with the VST.

4. The method as recited in Claim 3, wherein mapping the plurality of virtual sectors to at least one physical subsector through the Virtual Sector Table (VST) further includes providing a Sector Allocation Table (SAT) within the volatile memory, the SAT mapping the physical subsectors to the VST.

5. The method as recited in Claim 4, wherein providing a Sector Allocation Table (SAT) within the volatile memory further includes generating the SAT based at least on a unique group identifier that is stored in each physical subsector associated with storing the compressed data block.

6. The method as recited in Claim 5, wherein the Sector Allocation Table (SAT) is generated during a device initialization time.

7. The method as recited in Claim 1, wherein storing the compressed data block in at least one physical subsector within the non-volatile memory further includes associating each physical subsector with a unique group identifier.

8. The method as recited in Claim 7, wherein storing the compressed data block in at least one physical subsector within the non-volatile memory further includes writing each physical subsector associated with the compressed data block to the non-volatile memory in an a sequential order, but not necessarily a contiguous order.

9. The method as recited in Claim 1, wherein storing the compressed data block in at least one physical subsector within the non-volatile memory further includes associating a first physical subsector with at least one virtual sector identifier.

10. The method as recited in Claim 1, wherein storing the compressed data block in at least one physical subsector within the non-volatile memory further includes, maintaining input/output (I/O) operation status information within the non-volatile memory during on-going I/O operations.

11. The method as recited in Claim 1, wherein the non-volatile memory further includes a raw sector map configured to identify a plurality of reserved sectors within the non-volatile memory.

12. The method as recited in Claim 11, wherein the non-volatile memory further includes an input/output (I/O) operation status area that identifies the status of on-going I/O operations with respect to data stored within non-volatile memory.

13. The method as recited in Claim 11, wherein the non-volatile memory further includes a plurality of contiguously arranged reserved sectors.

14. The method as recited in Claim 11, wherein the non-volatile memory further includes a plurality of contiguously arranged physical subsectors.

15. The method as recited in Claim 14, wherein each of the physical subsectors includes a first portion that includes a group identifier.

16. The method as recited in Claim 14, wherein the group identifier identifies that the physical subsector is unused.

17. A computer-readable medium having computer-executable instructions for performing steps comprising:

receiving file system data;

storing the file system data in a plurality of reserved sectors within a non-volatile memory;

compressing the file system data stored within in the plurality of reserved sectors to create a compressed data block; and

storing the compressed data block in at least one physical subsector within the non-volatile memory.

18. The computer-readable medium as recited in Claim 17, wherein receiving file system data further includes presenting an operating system with a plurality of operatively accessible virtual sectors resulting in a virtual memory capacity that exceeds the actual physical capacity of the non-volatile memory.

19. The computer-readable medium as recited in Claim 18, wherein storing the compressed data block at least one physical subsector within the non-volatile memory further includes mapping the plurality of virtual sectors to at least one physical subsector through a Virtual Sector Table (VST) stored in a volatile memory and presenting the operating system with the VST.

20. The computer-readable medium as recited in Claim 19, wherein mapping the plurality of virtual sectors to at least one physical subsector through the Virtual Sector Table (VST) further includes providing a Sector Allocation Table (SAT) within the volatile memory, the SAT mapping the physical subsectors to the VST.

21. The computer-readable medium as recited in Claim 20, wherein providing a Sector Allocation Table (SAT) within the volatile memory further includes generating the SAT based at least on a unique group identifier that is stored in each physical subsector associated with storing the compressed data block.

22. The computer-readable medium as recited in Claim 21, wherein the Sector Allocation Table (SAT) is generated during a device initialization time.

23. The computer-readable medium as recited in Claim 17, wherein storing the compressed data block in at least one physical subsector within the non-volatile memory further includes associating each physical subsector with a unique group identifier.

24. The computer-readable medium as recited in Claim 23, wherein storing the compressed data block in at least one physical subsector within the non-volatile memory further includes writing each physical subsector associated with the compressed data block to the non-volatile memory in an a sequential order, but not necessarily a contiguous order.

25. The computer-readable medium as recited in Claim 17, wherein storing the compressed data block in at least one physical subsector within the non-volatile memory further includes associating a first physical subsector with at least one virtual sector identifier.

26. The computer-readable medium as recited in Claim 17, wherein storing the compressed data block in at least one physical subsector within the non-volatile memory further includes, maintaining input/output (I/O) operation status information within the non-volatile memory during on-going I/O operations.

27. The computer-readable medium as recited in Claim 17, wherein the non-volatile memory further includes a raw sector map configured to identify a plurality of reserved sectors within the non-volatile memory.

28. The computer-readable medium as recited in Claim 27, wherein the non-volatile memory further includes an input/output (I/O) operation status area that identifies the status of on-going I/O operations with respect to data stored within non-volatile memory.

29. The computer-readable medium as recited in Claim 27, wherein the non-volatile memory further includes a plurality of contiguously arranged reserved sectors.

30. The computer-readable medium as recited in Claim 27, wherein the non-volatile memory further includes a plurality of contiguously arranged physical subsectors.

31. The computer-readable medium as recited in Claim 30, wherein each of the physical subsectors includes a first portion that includes a group identifier.

32. The computer-readable medium as recited in Claim 30, wherein the group identifier identifies that the physical subsector is unused.

33. An arrangement for use in providing an application access a non-volatile memory, the arrangement comprising:

an operating system; and

a device driver, wherein the operating system is configured to exchange input/output (I/O) requests with the application and exchange corresponding file system requests with the device driver, and wherein the device driver is configured to store the file system data received from the operating system in a plurality of reserved sectors within the non-volatile memory, compress the file system data stored within in the plurality of reserved sectors to create a compressed data block, and store the compressed data block in at least one physical subsector within the non-volatile memory.

34. The arrangement as recited in Claim 33, wherein the device driver is further configured to present the operating system with a plurality of operatively accessible virtual sectors resulting in a virtual memory capacity that exceeds the actual physical capacity of the non-volatile memory.

35. The arrangement as recited in Claim 34, wherein the device driver is further configured to map the plurality of virtual sectors to at least one physical subsector through a Virtual Sector Table (VST) stored in a volatile memory and present the operating system with the VST.

36. The arrangement as recited in Claim 35, wherein the device driver is further configured to store a Sector Allocation Table (SAT) within the volatile memory, the SAT mapping the physical subsectors to the VST.

37. The arrangement as recited in Claim 36, wherein the device driver is further configured to generate the SAT based at least on a unique group identifier that is stored in each physical subsector associated with storing the compressed data block.

38. The arrangement as recited in Claim 37, wherein the device driver is further configured to generate the Sector Allocation Table (SAT) during a device initialization time.

39. The arrangement as recited in Claim 33, wherein the device driver is further configured to associate each physical subsector with a unique group identifier.

40. The arrangement as recited in Claim 39, wherein the device driver is further configured to write each physical subsector associated with the compressed data block to the non-volatile memory in an a sequential order, but not necessarily a contiguous order.

41. The arrangement as recited in Claim 33, wherein the device driver is further configured to associate a first physical subsector with at least one virtual sector identifier.

42. The arrangement as recited in Claim 33, wherein the device driver is further configured to maintain input/output (I/O) operation status information within the non-volatile memory during on-going I/O operations.

43. The arrangement as recited in Claim 33, further comprising:

a processor configured to run the operating system and the device driver; and

a non-volatile memory operatively coupled to the processor.

44. The arrangement as recited in Claim 43, wherein the processor and non-volatile memory are part of a set top box.

45. A non-volatile computer-readable medium having stored thereon a data structure, comprising:

raw sector map data;

a plurality of raw sectors as identified in the raw sector map that are configurable to store uncompressed data during input/output (I/O) operations;

I/O operation status data identifying on-going I/O operations; and

a plurality of subsectors contiguously arranged and configured to support the identified on-going I/O operations by storing compressed data blocks derived from the uncompressed data.

46. 47. The non-volatile computer-readable medium of Claim 45, wherein each of the plurality of subsectors further includes a group identifier that is uniquely associated with the associated compressed data block.